**实验三 HTTP响应对象**

一、实验目的

1. 了解和掌握如何从响应对象获得输出流对象；

2. 掌握如何设置响应类型；

3. 掌握如何设置响应头和状态码；

4. 重点掌握如何实现响应重定向。

二、实验原理

对Servlet的请求目的是要获得响应消息。Servlet要向用户发送响应数据首先要获得输出流对象。另外，通过响应对象还可向客户端发送状态码，也可以设置有关响应头，还可以对响应重定向。

三、实验内容与步骤

说明：Servlet中自行用注解描述url，自行决定如何访问，不用写web.xml文件。

1. **通过设置响应内容类型构建Excel电子表格（20分）**

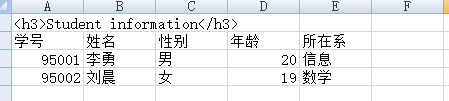


图3-1 ExcelServlet程序的运行结果

**（1）代码：**

package com.test1;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/zyh.xls")

public class ExcelServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.setContentType("application/vnd.ms-exce;charset=gb2312");

PrintWriter out = resp.getWriter();

out.print("<h3>Student information</h3>\n");

out.print("学号\t姓名\t性别\t年龄\t所在系\n");

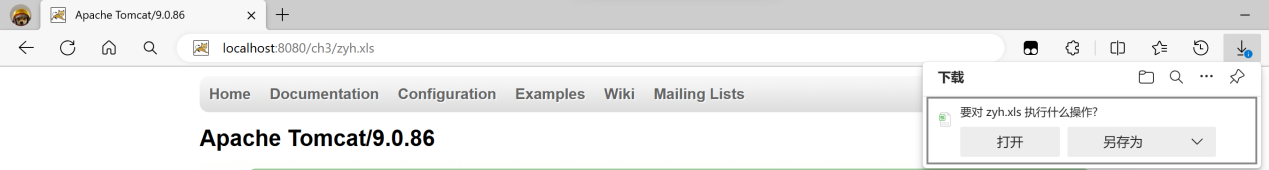
out.print("95001\t李勇\t男\t20\t信息\n");

out.print("95002\t刘晨\t女\t19\t数学\n");

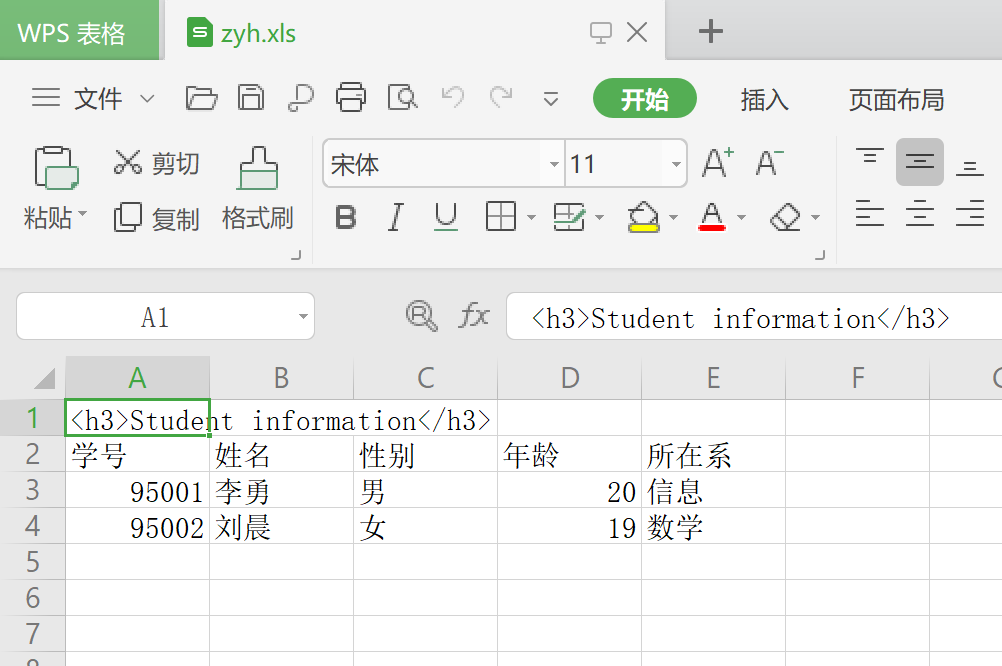
}

}

**（2）运行截图：**



访问url下载



Excel效果

1. **向客户发送二进制文件【可任意读取某个jar包文件】（20分）**

**（1）代码：**

package com.test2;

import java.io.\*;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.ServletOutputStream;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/servlet-api.jar")

public class FileServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.setContentType("application/jar;charset=utf-8");

String fileName = "servlet-api.jar";

ServletContext context = getServletContext();

String filePath = context.getRealPath("/WEB-INF/" + fileName);

File file = new File(filePath);

ServletOutputStream outStream = resp.getOutputStream();

try (InputStream in = new BufferedInputStream(new FileInputStream(filePath))) {

byte[] buffer = new byte[4096];

int bytesRead;

while ((bytesRead = in.read(buffer)) != -1) {

outStream.write(buffer, 0, bytesRead);

}

} finally {

outStream.flush();

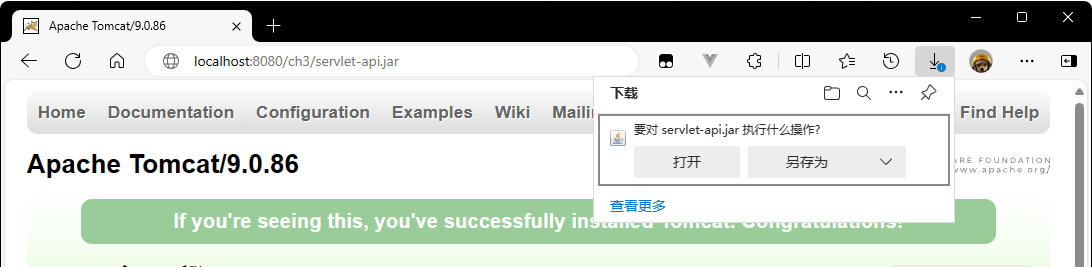
outStream.close();

}

}

}

**（2）运行截图：**



注意：此Servlet正确运行，需要将servlet-api.jar文件复制到C盘根目录下。

**3、实现ShowTimeServlet通过设置refresh响应头实现每2秒钟刷新一次页面。（20分）**

**（1）代码：**

package com.test3;

import java.io.IOException;

import java.io.PrintWriter;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/ShowTimeServlet")

public class ShowTimeServlet extends HttpServlet {

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.setContentType("text/html;charset=UTF-8");

PrintWriter out = resp.getWriter();

resp.setHeader("Refresh", "2");

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String time = dateFormat.format(new Date());

String name = "詹宇昊";

String Sno = "22023237";

PrintWriter out1 = resp.getWriter();

out1.println("<html><head><title>ShowTimeServlet</title></head>");

out1.println("<body>");

out1.println("<p>姓名：" + name + "</p>");

out1.println("<p>学号：" + Sno + "</p><br/>");

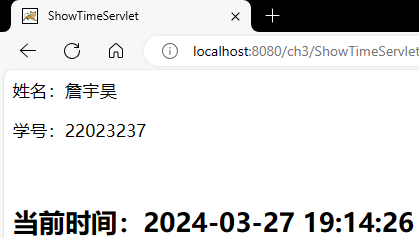
out1.println("<h2>当前时间：" + time + "</h2>");

out1.println("</body></html>");

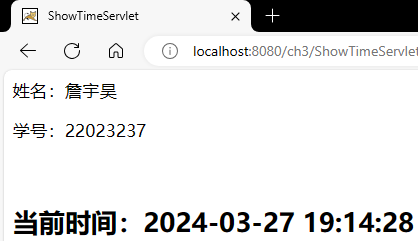
}

}

**（2）运行截图：**



刷新前



刷新后

**4、响应重定向（40分）**

【步骤1】建立login.html登录页面**（10分）**



图3-2 login.html登录页面运行效果

1. **代码：**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>login</title>

</head>

<body>

<form action="RedirectServlet" method="get">

<fieldset>

<legend>用户登录</legend>

<table cellspacing=10px>

<tr>

<td>用户名：</td>

<td><input type="text" name="username" /></td>

</tr>

<tr>

<td>密码：</td>

<td><input type="text" name="password" /></td>

</tr>

</table>

<input type="submit" value="登录" /> <input type="reset" value="取消" /><br />

</fieldset>

</form>

</body>

</html>

【步骤2】编写一个名为RedirectServlet的Servlet，在其中检索用户名和口令参数值，然后判断其是否正确（若用户名和口令都为admin即认为正确），如果正确将响应重定向到welcome.html页面，否则将响应重定向到error.html页面，这两个页面请你自己编写，存放在应用程序根目录。**（30分）**

1. **RedirectServlet代码：**

package com.test4;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/RedirectServlet")

public class RedirectServlet extends HttpServlet {

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

String username = req.getParameter("username");

String password = req.getParameter("password");

resp.setContentType("text/html;charset=UTF-8");

PrintWriter out = resp.getWriter();

out.println("<!DOCTYPE html>");

out.println("<html><body>");

if ("admin".equals(username) && "admin".equals(password)) {

req.setAttribute("username", username);

RequestDispatcher rd = req.getRequestDispatcher("/welcome.html");

rd.forward(req, resp);

} else {

RequestDispatcher rd = req.getRequestDispatcher("/error.html");

rd.forward(req, resp);

}

out.println("</body></html>");

}

}

**（3）代码：**

welcome.html代码如下：

<!DOCTYPE html>

<html>

<head>

<meta charset="GBK">

<title>welcome</title>

</head>

<body>

<h1>登陆成功！</h1>

<p id="showName"></p>

<script>

var url = new URLSearchParams(window.location.search);

var name = url.get('username');

document.getElementById('showName').innerText = name + ",欢迎回来";

</script>

</body>

</html>

error.html代码如下：

<!DOCTYPE html>

<html>

<head>

<meta charset="GBK">

<title>error</title>

</head>

<body>

<h1>输入错误！</h1>

<p>

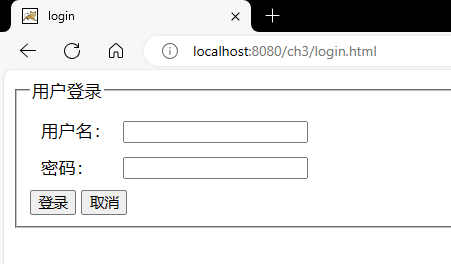
<a href='/ch3/login.html'>重新输入</a>

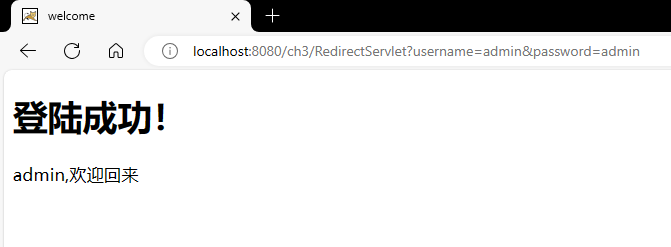
</p>

</body>

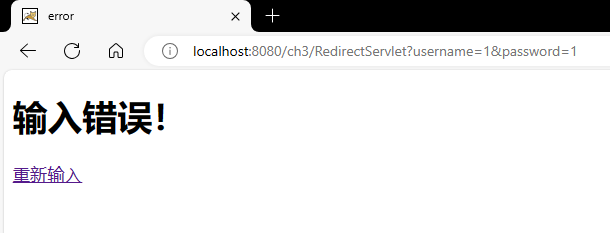
</html>

**（4）运行截图：**

****

login.html登录页面****

welcome.html成功页面

****

error.html失败页面